IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR U.S. LETTERS PATENT

Title:

PORTAL DESIGN SYSTEM AND METHODOLOGY

Inventors:

Mark S. Secrist
2012 Sheffield Court
Fort Collins, Colorado 80526
Citizenship: United States

Alex Nehmer
~~Geleener Strasse 30~~ Sofienstrasse 10
~~Boeblingen, Germany 71034~~ Tuebingen, Germany 72070
Citizenship: Germany

[illegible]

Rico Gundermann
Dorfstrase 20
Sulza, Germany 07751
Citizenship: Germany

PORTAL DESIGN SYSTEM AND METHODOLOGY

BACKGROUND

[0001] Over the last several years, web portals have risen in popularity as a way of aggregating, organizing, and presenting content in a highly uniform, customizable, and personalized way. As the technologies that enable the creation and management of these web portals have evolved, it is not only information content that is being offered, but application functionality as well. With application functionality making its way to the web portal, developers face a new dilemma in adapting application functionality to the characteristics and behavior of a web portal.

[0002] A portal is a web site that generally provides content and application functionality in a way that is both useful and meaningful to an end user. It may also serve some purpose from the portal provider's perspective, whether it is a public portal trying to attract web traffic to the site or an enterprise desiring to provide a central location for employees to obtain company information.

[0003] Most portals and portal frameworks contain the concept of a "portlet." A portlet is a window to a specific set of content within the overall context of the portal page. Many portlets support the ability to customize the information displayed within this window. From the perspective of the portal framework or platform, portlets tend to look and behave much the same as individual windows running in a MICROSOFT WINDOWS™-based operating system. Portlets may be minimized, maximized, and re-arranged around the display screen to suit the taste of the individual portal user. From the developer's perspective, a portlet is simply a piece of code that plugs into a generalized framework. Different portal frameworks implement the concept of a portlet differently. In some cases, the portlet is a collection of SUN MICROSYSTEM'S JAVA™ SERVER PAGES™ (JSP) pages. In other cases, it may be a special type of class that implements certain interfaces. Regardless of how it is implemented, the portlet is generally responsible for presenting a specific set of content that may be tailored to a user's preferences. The portal framework is responsible for handling the infrastructure services, such as providing the overall presentation, user management, security, and personalization.

[0004] In the design and implementation of portals, it is generally useful to understand that typical web applications do not necessarily map easily over to a portal paradigm, especially in applications with multi-page interaction. Therefore, when converting a typical multi-page web site into a portal, developers generally re-code the entire web site from scratch into the portal paradigm. This process takes up considerable development time and expense.

SUMMARY

[0005] Representative embodiments of the present invention are directed to a method for managing a portal adaptation lifecycle that may include determining a construction design for an adapted portal application, determining a model for separation or presentation logic and application logic of an existing Web application to be adapted into the portal application, determining a navigation construction for the adapted portal application, selecting a level of customization to apply to the adapted portal application, and selecting an isolation model for isolating business logic from the adapted portal application.

[0006] Further representative embodiments of the present invention are directed to a method for adapting a Web application to a portal application that may include adding at least one component of the Web application to a portal platform, creating a plurality of portlets within the portal platform, wherein each of the plurality includes pages representing a view for the at least one component of the Web application, defining at least one Web flow relationship representing interactions between the at least one component of the Web application, and converting the at least one Web flow relationship into at least one event, defined within the plurality of portlets, wherein the at least one event corresponds to the interactions.

[0007] Further representative embodiments of the present invention are directed to a methodology for converting a Web application into a portal application comprising moving Web components from the Web application into a portal framework corresponding to the portal application, dividing the portal application into a plurality of portlets, wherein each of the plurality serves content of one or more of the Web applications, and providing navigation resources to the portal application.

[0008] Further representative embodiments of the present invention are directed to a system for adapting a Web application to a portal application comprising means for adding one or more Web application components to the portal application, means for generating a plurality of portlets within the portal application, wherein each of the plurality includes a view for the one or more Web application components, means for defining at least one Web flow relationship representing interactions between the one or more Web application components, and means for converting the at least one Web flow relationship into at least one interaction event, defined within the plurality of portlets, wherein the at least one interaction event corresponds to the interactions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] For a more complete understanding of the present invention, reference is now made to the following descriptions taken in conjunction with the accompanying drawing, in which:

[0010] FIGURE 1 is a flow chart representing a design consideration lifecycle according to the teachings herein;

[0011] FIGURE 2 is block diagram illustrating a portal navigation relationship in one embodiment of the portalization methodology disclosed herein;

[0012] FIGURE 3 is a flowchart illustrating a methodology used in adapting a Web application to a portal using one embodiment of the portal creation system described herein;

[0013] FIGURE 4 is a flowchart illustrating a methodology used in adapting a Web application to a portal using steps in addition to those reflected in FIGURE 3;

[0014] FIGURE 5 is a flowchart illustrating a methodology for converting a Web application into a portal application according to an embodiment of portalization methodology disclosed herein;

[0015] FIGURE 6 is a flowchart illustrating additional steps that may be added to the methodology, illustrated in FIGURE 5, for converting a Web application into a portal application; and

[0016] FIGURE 7 is a flowchart of one embodiment of a method for managing a portal adaptation lifecycle.

DETAILED DESCRIPTION

[0017] FIGURE 1 is a flow chart representing design consideration lifecycle 10 according to the teachings herein. In designing Web applications that may either be implemented directly into a portal or converted from a standard hypertext markup language (HTML)-format into a portal format, the developer begins by considering, in step 100, any design constraints for the presentation and content of the site. Design constraints may include such issues as the visual theme or look 105 or the Web site or format 106 to be used in the presentation. In step 101, a model is generally created using elements such as flow charts, pseudo code, layout pages, and the like for developing the Web site having its presentation logic 107 separated from its application logic 108. In the HTML-format sites, developers typically hard-code formatting code along side application logic.

[0018] In step 102, the developer determines the appropriate site navigation model to employ by examining the model to find the experience that the developer has designed for the user. Navigation differs in portals because a call to a uniform resource locator (URL) from a portlet would cause the entire page to refresh, losing the portal in favor of the Web site called by the URL. The developer may consider implementing simple navigation through new window model 110, in which a URL call from the portlet would cause a new window to open for the called Web site. Additionally and alternatively, event call model 109 may be implemented, in which navigation occurs as an event call instead of a URL call. The URL calls or any Web component interactions may be converted into events for the portal application through appropriate code drafted by the developer. As the called site arrives at the portal, the code underlying the Web flow of the portal reformats the called Web page to be presented within the portlet from where the URL or event call originated.

[0019] Because portals generally allow users to customize or personalize content or layout of the Web site, the developer considers, in step 103, the desirable customization tools or models to include within the Web site. Customization tools and applications may be implemented by the developer by drafting the appropriate computer code that would use ad hoc editing model 111, in which a separate page, interactive window, JSP, or the like is run to gather and store information from the user accessing the Web site or portlet. For example, if a user desires to change content of a particular portlet, ad hoc editing model 111 would provide a separate page for the user to potentially select a different information source or add information to customize the portlet. A storage utility may be provided by the code drafted by the developer to preserve the customization information entered by the user Additionally and alternatively, existing data model 112 may be coded by the developer, which uses pre-existing information from the user's login to customize or personalize the Web site experience. For example, as a user logs in, the information that may be stored on the user is then accessible to the portal. If a user includes a game application in the portal that may require a new password or ID, the existing login information may be used to supply the new password and/or ID for the game application. In this process, the user is only required to log in once to the portal, while the portal logic takes care of logging into the game application for the user. In step 104, the developer may then select the different models for implementing the separation of application logic offered by the Web site, such as through model-view-control (MVC) applications 113 or Web services 114. Once these considerations have been addressed, the portal or convertible Web application may be constructed by the developer using the appropriate code instructions.

[0020] FIGURE 2 is block diagram illustrating portal navigation relationship 20 in one embodiment of the portalization methodology disclosed herein. Portal 200 may contain various and multiple portlets. The code making up portlet 201 causes certain information or logic to be presented to a viewer that includes navigation action 202. In a typical Web paradigm, navigation action 202 may take the form of a hyperlink to a URL. However, in the portal paradigm, should a hyperlinked URL be selected within portlet 201, portal 200 would be replaced by a new browser window displaying the Web site related to that URL, thus, losing portal 200. Therefore, the portal paradigm uses different methods for overcoming this limitation.

[0021] The embodiment illustrated in FIGURE 2 includes portal control 203. The code created by the developer underlying portal control 203 intercepts any hyperlinked URL calls made from portlet 201 or other locations within portal 200. Portal control 203 reconfigures or may even proxy the URL request of navigation action 202 from portlet 201. Based on that request, portal control 203 accesses Web page 205 or remote server 206 through Internet 204 to retrieve or receive information related to the request page or information. On receipt of Web page 205 or the information from remote server 206, portal control 203 configures and directs the information to be displayed within the construct of portlet 201. Portal control 203 would intercept all of the results from navigation requests made from portal 200 and direct the different results to the appropriate portlet for display.

[0022] FIGURE 3 is a flowchart illustrating methodology 30 used in adapting a Web application to a portal using one embodiment of the portal creation system described herein. Step 300 comprises adding at least one component of the Web application to a portal platform. Step 301 comprises creating a plurality of portlets within the portal platform, wherein each of the plurality includes pages representing a view for the at least one component of the Web application. Step 302 comprises defining at least one Web flow relationship representing interactions between the at least one component of the Web application. Step 303 comprises converting the at least one Web flow relationship into at least one event, defined within the plurality of portlets, wherein the at least one event corresponds to the interactions.

[0023] For example, an existing HTML page may include several different components that provide data to present to a user over the Web browser. These components may provide various information, such as news, access to email, stock prices, and the like, that is gathered from a database, Web service, or the like. A developer would generally move the components over to a portal platform. For each of the components moved, the developer would write code for a portlet. The portlets provide the display space for the information. As described above with respect to FIGURE 2, the developer would then select and write code to handle the Web flow for interactions within the portlets and then convert the particular Web flow element to at least one event that corresponds to the interactions that may occur within the portlet, such as clicking to check email, clicking to read a news story, and the like.

[0024] In other embodiments, the methodology of FIGURE 3 may include further steps, as illustrated in FIGURE 4. Depending on the application or content being offered by the portal, the developer may select the level of customization or personalization to allow in the portal and implement the selected customization through appropriate computer code. In step 400, a customization application is coded by the developer to interact with the user to obtain customization information. This information may then be stored in step 402 by defining a storage utility in the appropriate underlying computer code for storing the information obtained from the user in the portal framework. This information may then be used by the portal code logic in assembling the presentation layers of the portal information. Additionally or alternatively, a personalization application may be coded by the developer in step 401 that retrieves existing login information related to a user. Once retrieved, this personalization information may also be incorporated into the presentation logic to make the user interface personalized to the user.

[0025] Additionally or alternatively, a developer may select a certain level or procedure to separate the business logic end from the presentation logic end and implement the selected separation in the code structure of the portal. In one embodiment illustrated by step 403, the business logic is modified to return its output as extensible markup language (XML) documents. Modern portal frameworks include utilities to read and present XML documents using template files such as extensible stylesheet language transformation (XSLT) documents. Thus, by returning XML documents, the output of the business logic becomes a standard format recognizable by most portal frameworks.

[0026] Additionally or alternatively to the modification of the existing business logic, client components may be created using appropriate code designed by the developer in step 404 for locating Web services to provide the business logic to the portal. Again, Web service or client interfaces may be provided in Simple Object Access Protocol (SOAP), which is an XML-based protocol, but leverages a standardized means for integrating Web services into the portal system. Web services may reside anywhere or be provided by different sources. Therefore, the portal system receives a good deal of application logic separation by using Web services to provide the business logic end.

[0027] Methodology 50 for converting a Web application into a portal application is illustrated in FIGURE 5. A Web application is typically constructed from a number of Web components appropriately coded by the developer that provide some information or application logic to the user. In step 500, the code underlying the Web components are moved by the developer from the Web application into a portal framework corresponding to the portal application. The portal application is then divided by the developer added code into a plurality of portlets in step 501, wherein each of the plurality serves content of one or more or the Web applications. Dividing the portal application into a plurality of portlets allows the componentization of the Web application to be realized in the portal framework. Because the navigation scheme of a Web application works from a different underlying paradigm, navigation resources are then provided to the portal application in step 502 by the developer adding the appropriate navigation code to the portal application.

[0028] In other embodiments, methodology 50 of FIGURE 5 may include further steps 60, as illustrated in FIGURE 6. In step 600, a construction layout is designed for the plurality of portlets responsive to either a visual theme of the portal application or content formatting of the portal application. The developer and any graphics artist may then code the construction layout to implement the desired usual theme. In step 601, a certain level of customization is selected to apply to the portal application. The customization code written by the developer allows users to change the presentation of the portal to his or her own desire. Process logic is then isolated from the portal application in step 602. This may be accomplished by, for example, including underlying code designed for modifying the process logic to output at least one data-descriptive metal language or by implementing process logic through use of one or more Web services. The navigation resources provided in step 502 may be implemented, in step 603, by drafting computer code to convert uniform resource locator (URL) calls from the Web application to interaction events for the portal application, or by including computer code to create a new window, in step 604, for information retrieved in response to a call to the URL from the Web application.

[0029] FIGURE 7 is a flowchart of one embodiment of a method for managing a portal adaptation lifecycle. Step 700 comprises determining a construction design for an adapted portal application. Step 701 comprises determining a model for separation or presentation logic and application logic of an existing Web application to be adapted into the portal application. Step 702 comprises determining a navigation construction for the adapted portal application. Step 703 comprises selecting a level of customization to apply to the adapted portal application. Step 704 comprises selecting an isolation model for isolating business logic from the adapted portal application.